

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently amended) A method, comprising:

determining a number of conflicting entries in a ~~first~~ software redirection table having a first set of entries, wherein the first set of entries is capable of being mapped to a second set of entries of a ~~second~~ hardware redirection table; and

mapping the first set of entries to the second set of entries, based on the number of conflicting entries in the ~~first~~ software redirection table, wherein the conflicting entries are caused if at least two entries of the software redirection table that are capable of being mapped to one entry of the hardware redirection table indicate different receive queues, the method further comprising:

(a) determining whether the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table, wherein the number of conflicting entries are determined in response to determining that the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table; and

(b) indicating that packets associated with conflicting entries are to be directed to one receive queue, in response to determining that the number of conflicting entries is less than a threshold.

2. (Canceled)

3. (Currently amended) The method of claim [[2]] 1, wherein the one receive queue is a first receive queue, the method further comprising:

distributing those packets that are in the [[one]] first receive queue among all processors of a plurality of processors for processing; and

~~processing packets in other receive queues in different processors.~~

processing those packets that are in a second receive queue in a first processor of the plurality of processors; and

processing those packets that are in a third receive queue in a second processor of the plurality of processors.

4. (Currently amended) The method of claim [[2]] 1, further comprising:
indicating that all packets are to be directed to a single receive queue, in response to determining that the number of conflicting entries is not less than the threshold.

5. (Original) The method of claim 4, further comprising:
processing receive side scaling in software, wherein processing receive side scaling further comprises creating virtual queues and queuing deferred procedure calls to corresponding processors via a device driver.

6. (Currently amended) The method of claim [[2]] 1, further comprising:
programming the hardware redirection table in accordance with the software redirection table, in response to determining that the first set of entries in the software redirection table does not have more members than the second set of entries in the hardware redirection table.

7. (Original) The method of claim 1, wherein determining and mapping are performed by a device driver in a computational platform having a plurality of processors.

8. (Currently amended) The method of claim 1, wherein the ~~first~~ software redirection table is associated with an operating system that supports receive side scaling, wherein the ~~second~~ hardware redirection table is implemented in a hardware device coupled to a computational platform having a plurality of processors, and wherein the ~~second~~ hardware redirection table is of a fixed size.

9. (Currently amended) A system, comprising:
at least one processor;
a network interface coupled to the at least one processor; and

program logic including code that is capable of causing the at least one processor to be operable to:

(i) determine a number of conflicting entries in a ~~first~~ software redirection having a first set of entries, wherein the first set of entries is capable of being mapped to a second set of entries of a ~~second~~ hardware redirection table implemented in the network interface; and

(ii) map the first set of entries to the second set of entries, based on the number of conflicting entries in the ~~first~~ software redirection table, wherein the conflicting entries are caused if at least two entries of the software redirection table that are capable of being mapped to one entry of the hardware redirection table indicate different receive queues, wherein the program logic is further capable of causing the at least one processor to be operable to:

(a) determine whether the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table, wherein the number of conflicting entries are determined in response to a determination that the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table; and

(b) indicate that packets associated with conflicting entries are to be directed to one receive queue, if the number of conflicting entries is less than a threshold.

10. (Canceled)

11. (Currently amended) The system of claim ~~[[10]]~~ 9, wherein the one receive queue is a first receive queue, wherein the program logic is further capable of causing the at least one processor to be operable to:

distribute those packets that are in the ~~[[one]]~~ first receive queue among all processors of a plurality of processors for processing; and

~~process packets in other receive queues in different processors.~~

process those packets that are in a second receive queue in a first processor of the plurality of processors; and

process those packets that are in a third receive queue in a second processor of the plurality of processors.

12. (Currently amended) The system of claim [[10]] 9, wherein the program logic is further capable of causing the at least one processor to be operable to:

indicate that all packets are to be directed to a single receive queue, if the number of conflicting entries is not less than the threshold

13. (Original) The system of claim 12, further comprising:

a device driver, wherein the device driver is operable to process receive side scaling in software by creation of virtual queues, and wherein the device driver is capable of queuing deferred procedure calls associated with the virtual queues to corresponding processors.

14. (Currently amended) The system of claim [[10]] 9, wherein the program logic is further capable of causing the at least one processor to be operable to:

program the hardware redirection table in accordance with the software redirection table, in response to the determination that the first set of entries in the software redirection table does not have more members than the second set of entries in the hardware redirection table.

15. (Original) The system of claim 9, further comprising:

a device driver operable to determine the number of conflicting entries and map the first set of entries.

16. (Currently amended) The system of claim 9, wherein the ~~first~~ software redirection table is associated with an operating system that supports receive side scaling, wherein the ~~second~~ hardware redirection table is implemented in the network interface, and wherein the ~~second~~ hardware redirection table is of a fixed size.

17. (Currently amended) A system, comprising:

a computational platform;

a storage controller implemented in the computational platform;
at least one processor coupled to the computational platform;
a network interface coupled to computational platform; and
program logic including code that is capable of causing the at least one processor to be operable to:

(i) determine a number of conflicting entries in a ~~first~~ software redirection table having a first set of entries, wherein the first set of entries is capable of being mapped to a second set of entries of a ~~second~~ hardware redirection table, wherein the ~~second~~ hardware redirection table is implemented in the network interface; and

(ii) map the first set of entries to the second set of entries, based on the number of conflicting entries in the ~~first~~ software redirection table, wherein the conflicting entries are caused if at least two entries of the software redirection table that are capable of being mapped to one entry of the hardware redirection table indicate different receive queues, wherein the program logic is further capable of causing the at least one processor to be operable to:

(a) determine whether the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table, wherein the number of conflicting entries are determined in response to a determination that the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table; and

(b) indicate that packets associated with conflicting entries are to be directed to one receive queue, if the number of conflicting entries is less than a threshold.

18. (Canceled)

19. (Currently amended) The system of claim ~~[[18]]~~ 17, wherein the one receive queue is a first receive queue, wherein the program logic is further capable of causing the at least one processor to be operable to:

distribute those packets that are in the ~~[[one]]~~ first receive queue among all processors of a plurality of processors for processing; and

~~process packets in other receive queues in different processors.~~
process those packets that are in a second receive queue in a first processor of the plurality of processors; and
process those packets that are in a third receive queue in a second processor of the plurality of processors.

20.(Currently amended) The system of claim [[18]] 17, wherein the program logic is further capable of causing the at least one processor to be operable to:

indicate that all packets are to be directed to a single receive queue, in response to the determination that the number of conflicting entries is not less than the threshold.

21. (Currently amended) An article of manufacture, comprising a storage medium having stored therein instructions that are operable by a machine to:

determine a number of conflicting entries in a ~~first~~ software redirection table having a first set of entries, wherein the first set of entries is capable of being mapped to a second set of entries of a ~~second~~ hardware redirection table; and

map the first set of entries to the second set of entries, based on the number of conflicting entries in the ~~first~~ software redirection table, and wherein the conflicting entries are caused if at least two entries of the software redirection table that are capable of being mapped to one entry of the hardware redirection table indicate different receive queues, wherein the instructions are further operable by a machine to:

(a) determine whether the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table, wherein the number of conflicting entries are determined in response to determining that the first set of entries in the software redirection table has more members than the second set of entries in the hardware redirection table; and

(b) indicate that packets associated with conflicting entries are to be directed to one receive queue, in response to determining that the number of conflicting entries is less than a threshold.

22. (Canceled)

23. (Currently amended) The article of manufacture of claim [[22]] 21, , wherein the one receive queue is a first receive queue, wherein the instructions are further operable by a machine to:

distribute those packets that are in the [[one]] first receive queue among all processors of a plurality of processors for processing; ~~and~~

~~process packets in other receive queues in different processors.~~

process those packets that are in a second receive queue in a first processor of the plurality of processors; and

process those packets that are in a third receive queue in a second processor of the plurality of processors.

24. (Currently amended) The article of manufacture of claim [[22]] 21, wherein the instructions are further operable by a machine to:

indicate that all packets are to be directed to a single receive queue, in response to determining that the number of conflicting entries is not less than the threshold.

25. (Original) The article of manufacture of claim 24, wherein the instructions are further operable by a machine to:

process receive side scaling in by creation of virtual queues, wherein a device driver is capable of queuing deferred procedure calls associated with the virtual queues to corresponding processors.

26. (Currently amended) The article of manufacture of claim [[22]] 21, wherein the instructions are further operable by a machine to:

program the hardware redirection table in accordance with the software redirection table, in response to determining that the first set of entries in the software redirection table does not have more members than the second set of entries in the hardware redirection table.

27. (Original) The article of manufacture of claim 21, wherein determination of the number of conflicting entries and mapping the first set of entries are performed by a device driver in a computational platform having a plurality of processors.

28. (Currently amended) The article of manufacture of claim 21, wherein the ~~first~~ software redirection table is associated with an operating system that supports receive side scaling, wherein the ~~second~~ hardware redirection table is implemented in the network interface, and wherein the ~~second~~ hardware redirection table is of a fixed size.